

Affect Product

Dyson Pure Hot+Cool Air Purifier

Impact

Escalation of Privileges, Remote control

Threat Model

This report assumes an unprivileged attacker can purchase a Dyson device and register a user account as legitimate users can do. The adversary can reverse engineer his own Dyson mobile app and app traffic.

Consider the scenario where the usage rights (or ownership) of the device are transferred from one user to the next. Such a transfer is common in IoT lifecycle; for example, a prior employee or prior apartment/Airbnb/hotel tenant whose permission has been revoked still aims to operate the Dyson devices.

The Security Vulnerability

Any Dyson user account can publish to MQTT topics of other users' devices, such as `"527/{DeviceSerialNumber}/command"`.

End-to-End Attacks:

By exploiting the weaknesses above, this report describes a practical attack with technical details as follows:

1. The attacker downloads the Dyson app and registers an account.
2. Once the attacker (e.g., an ex-employee or tenant/guest, see the threat model) has temporary permission to access a Dyson device (a Dyson Purifier in our experiment), he can inspect the traffic of his Dyson app and obtain the cognito credential of his own account (including `"aws_access_key_id"`, `"aws_secret_access_key"` and `"aws_session_token"`) and the device's serial number(as part of device control topic).
3. Later, the admin (or a new user of the Dyson device) resets the device or revokes the prior user's (attacker) permission. Then, in the Dyson app, the attacker can not see the device.
4. However, the attacker can develop a simple script, which uses the above credentials and connects to Dyson's AWS IoT endpoint `"a1u2wvl3e2lrc4-ats.iot.eu-west-1.amazonaws.com"` just like the Dyson app can do. But the script is under the attacker's control. The script publishes MQTT messages (including commands) to MQTT topics like `"527/{serial number}/command"` (including the target device's serial number), effectively controlling the target Dyson devices. And, the script can also subscribe to MQTT topics like `"527/{serial number}/status/current"` to monitor the device status.